

Software Engineering

3de BAC Informatica (Computer Science)
[Academic year 2023-2024]

© Prof. Serge Demeyer

(This slide deck is significantly revised compared to previous editions.
Extra - changed - new info is marked with variants of ****New**** ****Revised****)



**Universiteit
Antwerpen**

Opleiding Informatica — Doel en Ambitie

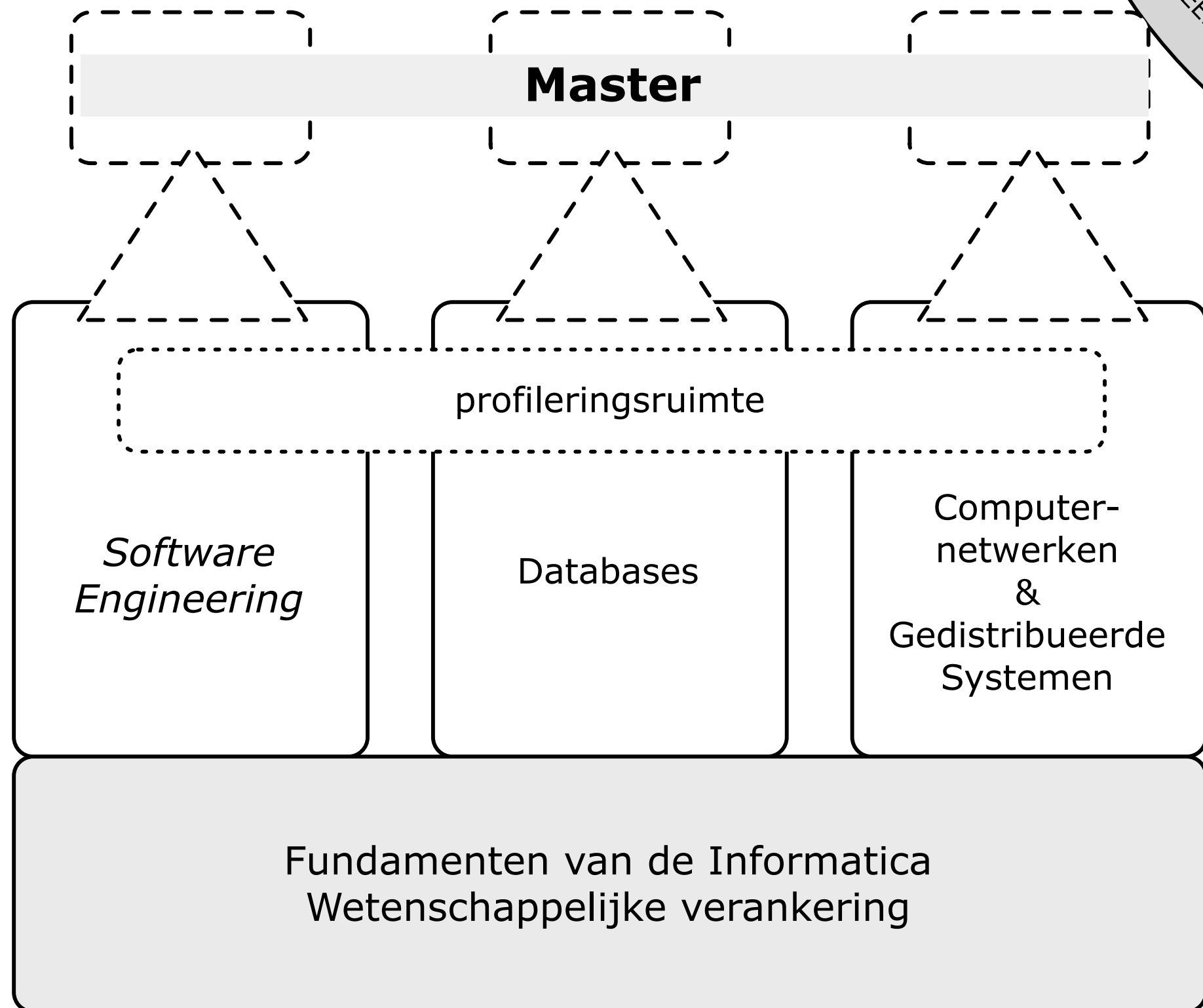
Doelstelling: Het doel van de opleiding Informatica aan de Universiteit Antwerpen is bekwame en wetenschappelijk gevormde informatici af te leveren. Na hun opleiding

- (a) zijn zij in staat zich de nieuwste technologische ontwikkelingen in de diverse deelgebieden eigen te maken,
- (b) kunnen zij deze waar nodig ook exploiteren binnen hun beroepscontext, en
- (c) kunnen zij zelf een originele bijdrage leveren tot de verdere evolutie van de informatica.

Ambitie: De onderwijscommissie Informatica van de Universiteit Antwerpen wil een unieke opleiding aanbieden gebaseerd op de aanbevelingen van gerenommeerde wetenschappelijke organisaties zoals ACM, IEEE en SIAM. Ze wenst dat haar alumni voldoende diepgang en flexibiliteit bezitten om gegeerd te zijn op de arbeidsmarkt en academische posities te bekleden in universiteiten en onderzoeksinstellingen met wereldfaam. Daartoe hanteert zij een transparant en activerend leerproces, conform de visie van de Universiteit Antwerpen rond “studentgecentreerd onderwijs”.

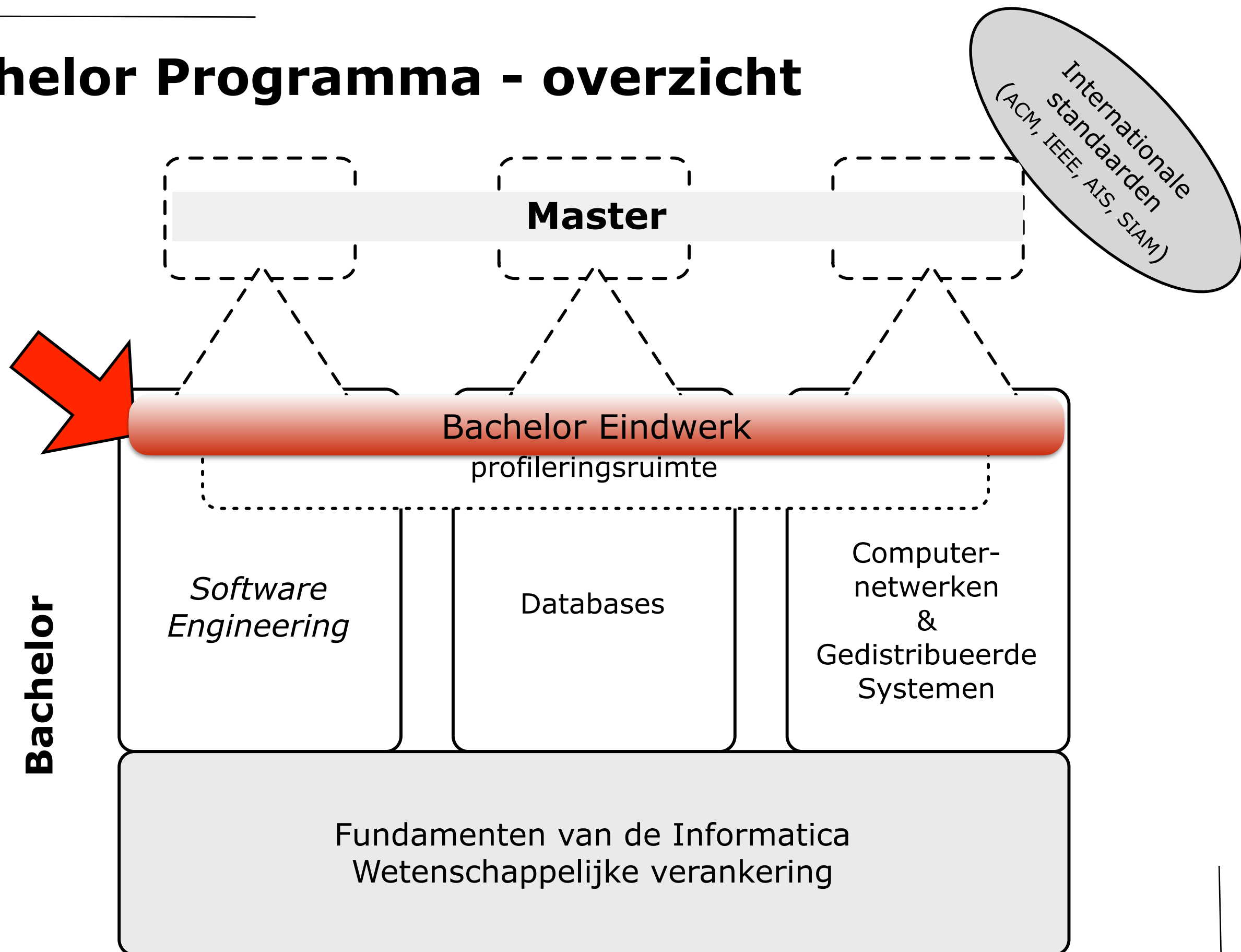
Bachelor Programma - overzicht

Bachelor



Internationale
standaarden
(ACM, IEEE, AIS, SIAM)

Bachelor Programma - overzicht



Bachelor Eindwerk



Bachelor eindwerk

Studiegidsnr:	1002WETBAE
Vakgebied:	Informatica
Academiejaar:	2019-2020
Semester:	2e semester
Inschrijvingsvereisten:	1) MINIMUM 8/20 voor Gevorderd programmeren en Programming project databases EN 2) INGESCHREVEN (of credit behaald) voor Software engineering Telecommunicatiesystemen, Gedistribueerde systemen, Artificial intelligence en Compilers.
Contacturen:	160
Studiepunten:	12
Studiebelasting:	336
Contractrestrictie(s):	Niet te volgen onder examencontracten
Credit vereist voor behalen diploma:	Credit vereist: Ja
Instructietaal:	Nederlands
Examen:	2e semester
Mogelijkheid 2de zitting:	Neen
Lesgever(s)	Steven Latré

Capstone

Kerncompetenties Software Engineering

Bachelor = bekwame informaticus

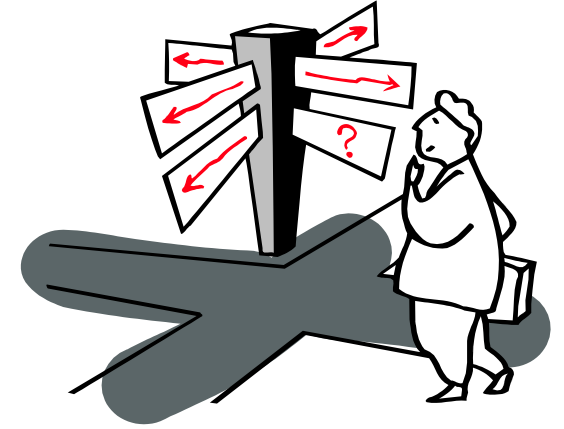
Analyse en ontwerp voor kleinschalige software projecten	+
Implementatie van nieuwe softwaresystemen	+
Onderhoud van bestaande softwaresystemen	++
Implementatie en onderhoud van een databank	
Beheer van een lokaal netwerk	
Vakbekwaamheid	++
Maatschappij	+++
Communicatievaardigheden	+

Academische Bachelor = wetenschappelijke vorming

Wiskundige basis	
Formeel denken en abstraherend vermogen	+
Levenslang leren	++
Wetenschappelijke aanpak	+
Wetenschappelijke basis	
Autonoom en creatief functioneren	+

HOOFDSTUK 0 – Praktische Zaken

- Doel
 - + Professionele Informaticus
 - + Plaats in het Curriculum
 - Kerncompetenties
 - + Beoordelingscriteria
 - + Examen
 - + Voorbeeldvragen
- Literatuur
- Inhoudstafel



Doel

- Programmaboekje
 - + “Het doel bestaat erin om de student een brede basis te verschaffen in het bouwen van softwaresystemen die te complex zijn om door één persoon gerealiseerd te worden.”
 - + Dus
 - brede basis
 - > véél technieken
 - complexe systemen
 - > schaalbare technieken

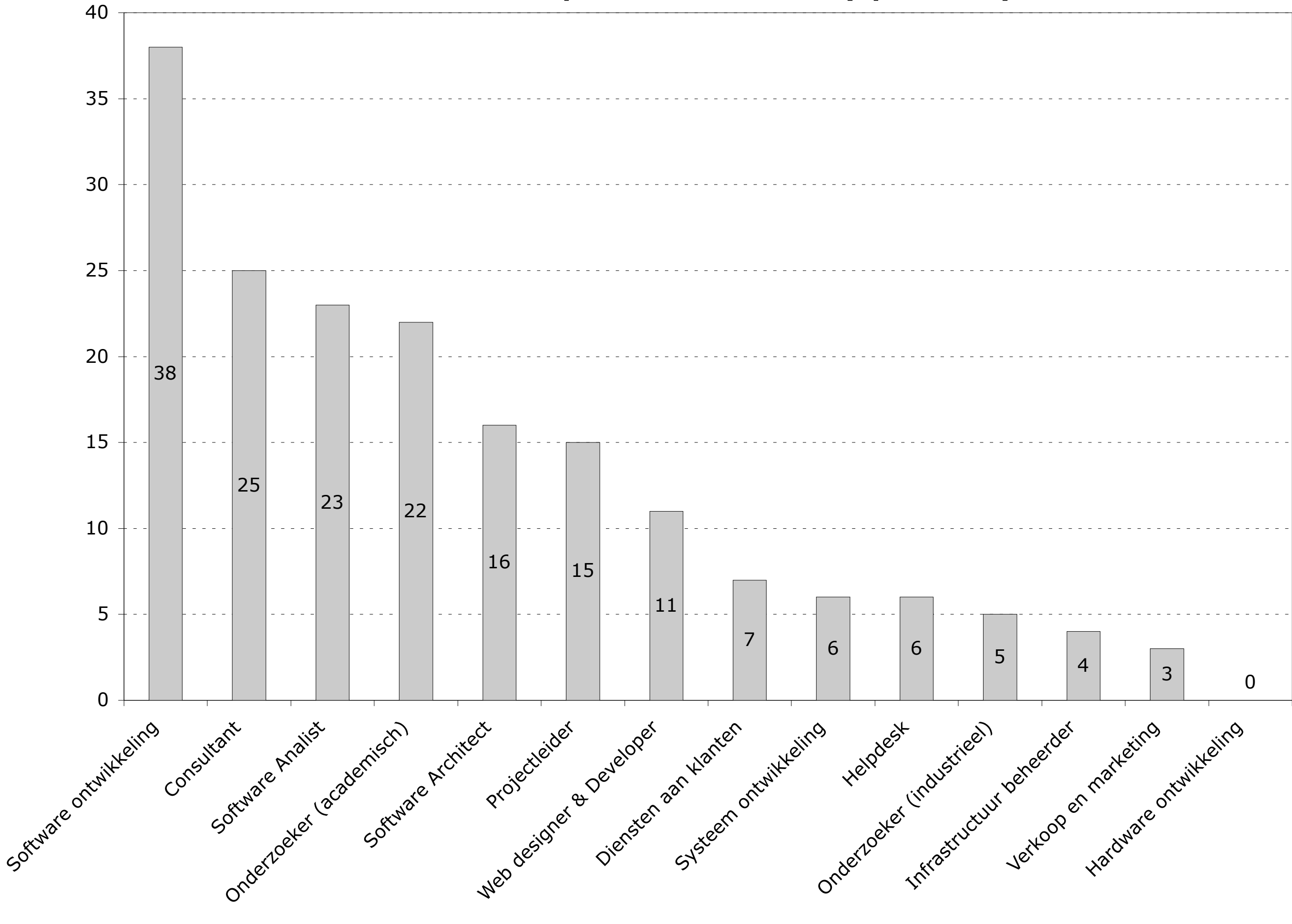
Professionele Informaticus

- diverse domeinen
 - + traditionele “data processing”
(banken, verzekeringen)
 - + spitstechnologie
(cloud, AI, cyber-physical)
 - + telecommunicatie
(netwerkbeheer, e-commerce)
 - + ...
 - + onderzoek
- zij verwachten
 - + technische virtuozen
 - diverse specialiteiten
 - + groepsspelers
 - sociale vaardigheden

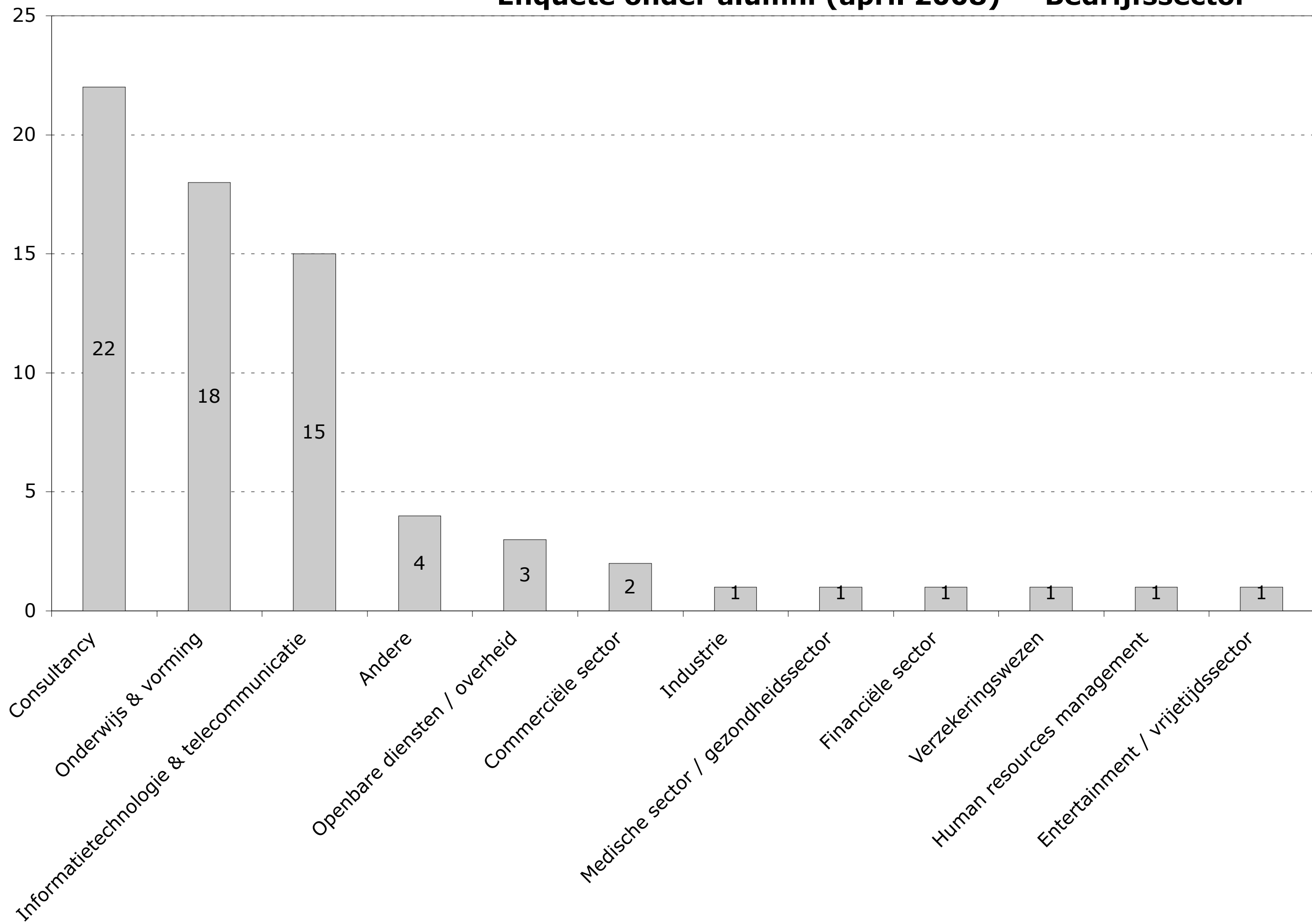


Public Domain

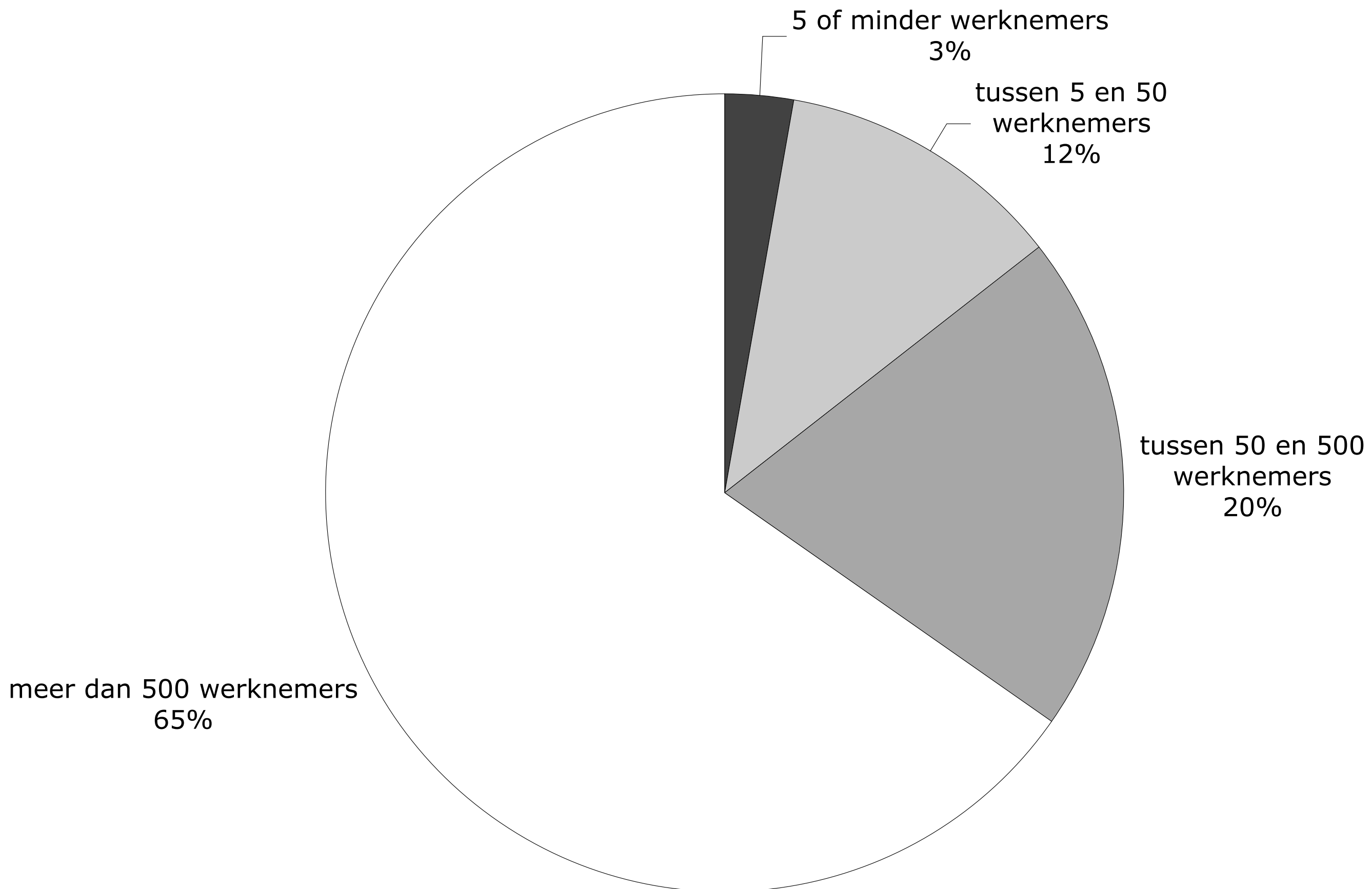
Enquête onder alumni (april 2008) — JobProfiel



Enquête onder alumni (april 2008) — Bedrijfssector

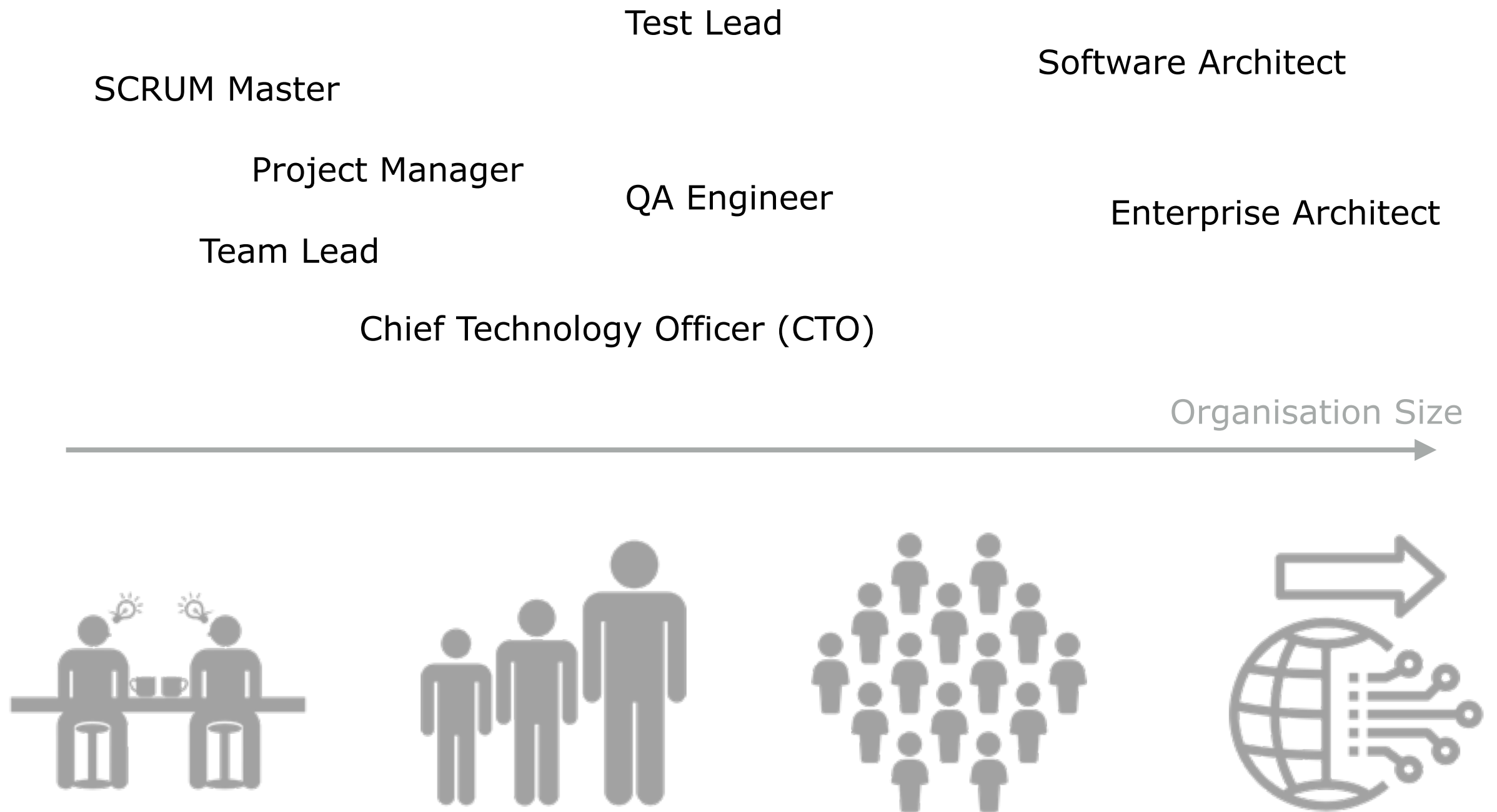


Enquête onder alumni (april 2008) — Bedrijfsgrootte



Job Profiles (Software Engineering)

****New****



Criteria - Selectie

- Accuraatheid
 - + Een professionele software engineer werkt in groep en moet dus op een accurate manier kunnen communiceren met zijn collega's en eindgebruikers.
 - * Juist gebruik van terminologie
 - * Parate kennis definities
- Toepasbare kennis
 - + Een professionele software engineer moet in staat zijn gekende technieken toe te passen in een variërende context.
 - * "Know-how"
 - * Oefeningen

SELECTIE

- Je *moet* dit kunnen demonstreren tijdens het examen om te slagen!



Criteria - Diversificatie

- Inzicht
 - + Een professionele software engineer moet technische keuzes kunnen verantwoorden
 - * "Know when"
 - * Afwegingen maken

DIVERSIFICATIE:

- De mate waarin je dit kunt demonstreren tijdens het examen laat je toe je te onderscheiden van je collega's.



Tussentijdse Opdrachten

****Herzien****

- (Bijna) wekelijks
- Oefeningen: toepassing van de theorie van de week.
 - \Rightarrow Rapporteren via mondelinge presentatie (om de 2 à 3 weken)
- In groep: 3 personen.
- Kwalitatieve feedback \Rightarrow evolutie is belangrijk.

Evaluatie

- *Quotering volgens gewicht*
 - 30% - *Mijlpaal 1 [Requirements + Architecture + Project Management]*
 - 50% - *Mijlpaal 2 [Testing + Formal Specifications + State Charts]*
 - 20% - *Mijlpaal 3 [Refactoring + Quality Control]*

Een resultaat > 12 op 20 is noodzakelijk om te slagen voor dit vak!

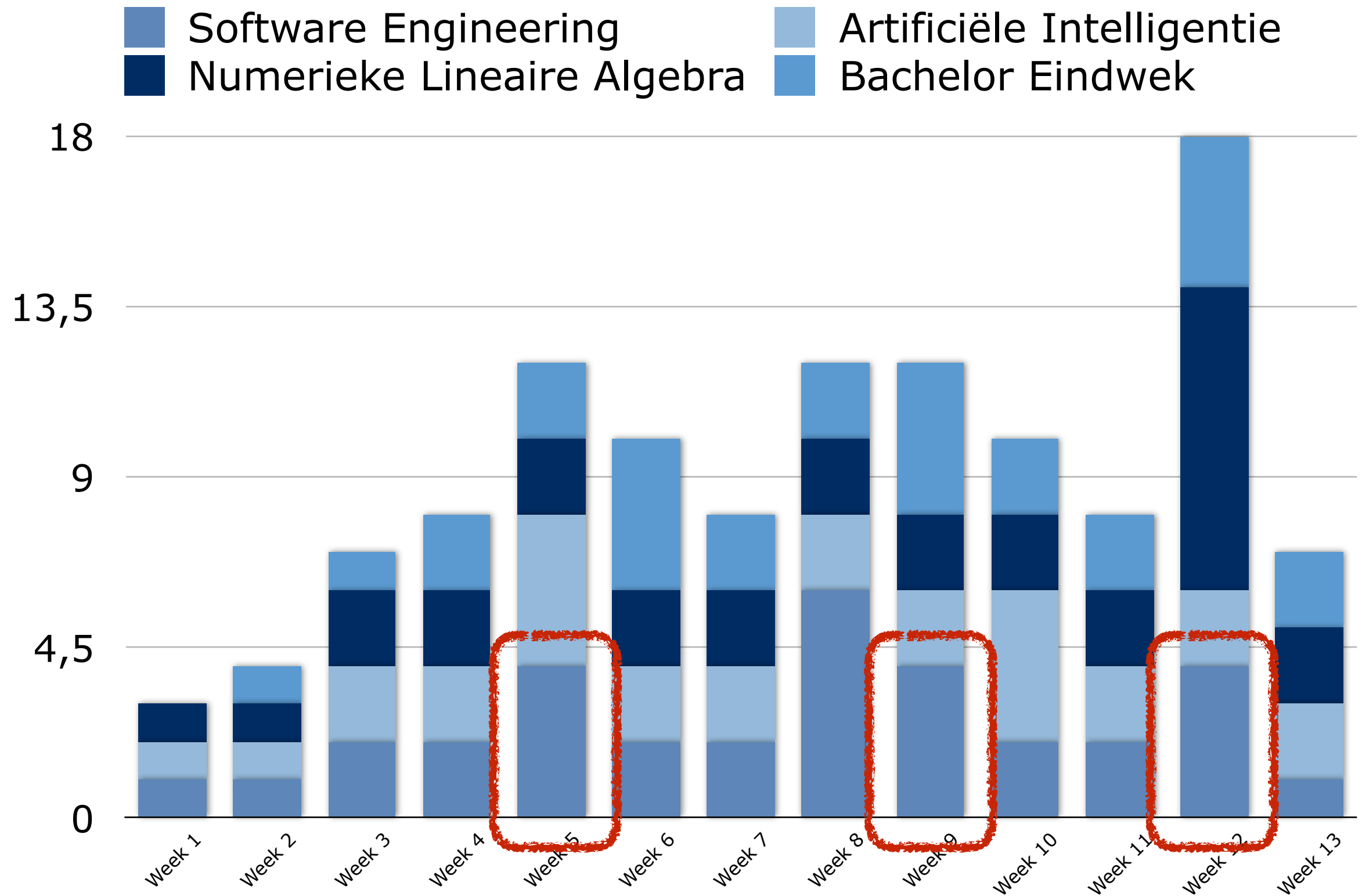
Ideale Semester Planning

****Herzien****

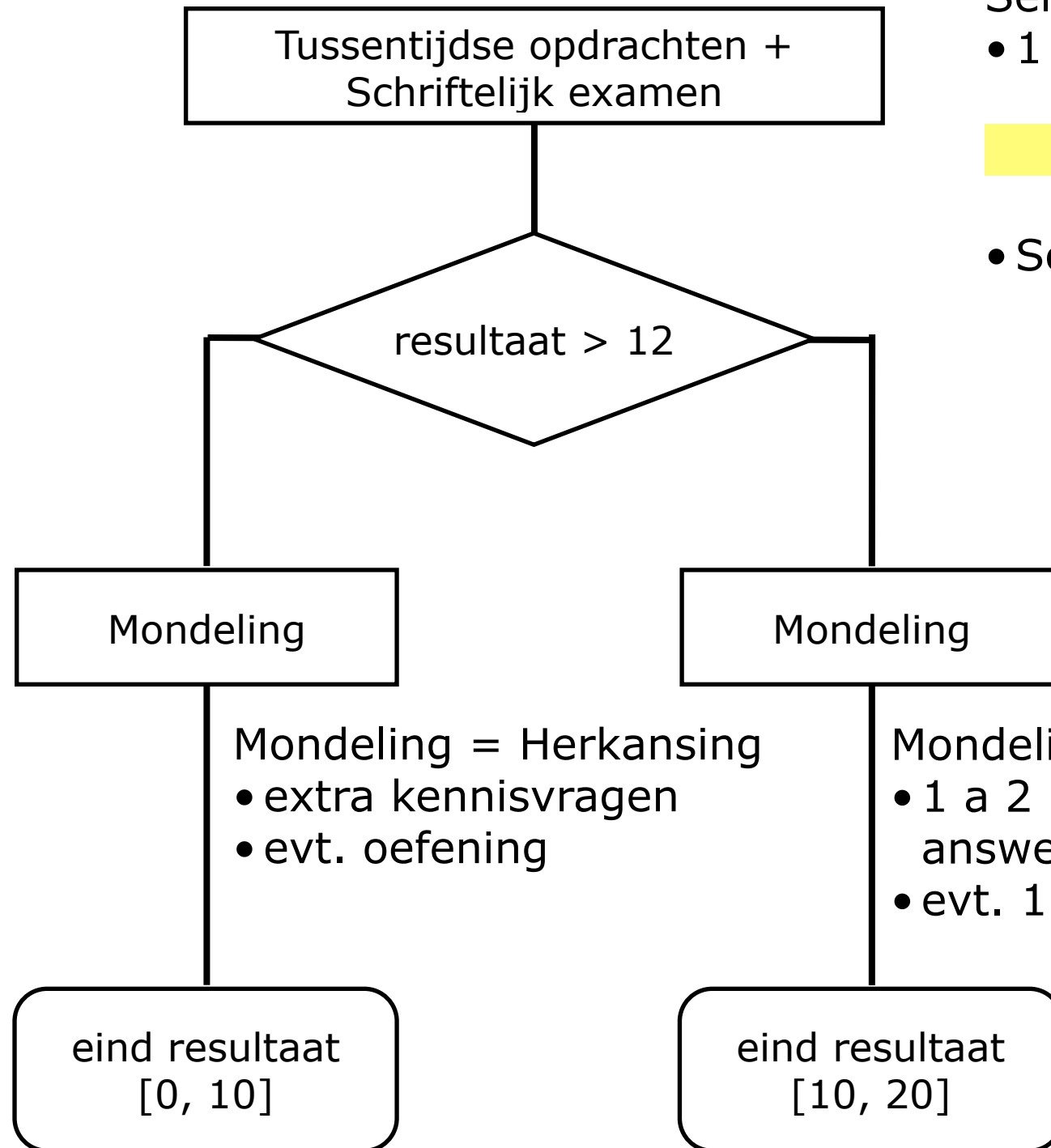
week 1	[T] Introduction	-- geen oefeningen
week 2	[T] Requirements	[P] Requirements
week 3	[T] Software Architecture	[P] Software Architecture
week 4	[T] Project Management	[P] Project Management
week 5	[T] Design by Contract	-- EVALUATIE
week 6	[T] Testing	[P] Testing
week 7	[T] Formal Specifications	[P] Formal Specifications
week 8	[T] Domain Models	[P] State Charts
week 9	[T] Software Quality	-- EVALUATIE
week 10	[T] Software Metrics	[P] Refactoring
week 11	[T] Refactoring	[P] Software Quality
week 12	(-- reserve)	-- EVALUATIE
week 13	[T] Conclusion	-- geen oefeningen

<https://ansymore.uantwerpen.be/software-engineering-3e-bac/tijdsschema>

Uitstellen?



Examen



Selectie (minimumnorm)

- 1 tussentijdse opdracht per hoofdstuk
 - Tijdens het jaar op te leveren
 - *Presentatie + Kwalitatieve feedback*
 - Resultaat > 12
- Schriftelijk examen
 - 1 kennis vraag per hoofdstuk
 - (cfr. "You should know the answer to these questions")
 - beperkte oefeningen
 - Resultaat > 12

Voorbeeldvragen (schriftelijk)

Enkele voorbeeldvragen

1. Geef 2 redenen waarom het waterval model onrealistisch is.
2. "Het systeem is voor 93% correct" is een geldige uitspraak.
Ja / Neen
Waarom?
3. Bij het overschrijven van een methode in een subklasse ...
 - (a) moet de preconditionie gelijk blijven
 - (b) mag de preconditionie zwakker worden
 - (c) mag de preconditionie sterker wordenWaarom?

Modelantwoorden

1.
gebruikers kunnen behoeften nooit volledig specificeren
een werkende versie is veel te laat beschikbaar
2. Ja / Neen
Correctheid is een absolute eigenschap.
3. (b)
Een subklasse moet minstens hetzelfde contract vervullen.

Voorbeeldvragen (mondeling)

Een voorbeeld v. e. inzichtsvraag

+ Leveren CRC-kaarten het best mogelijke ontwerp? Argumenteer?

+ Modelantwoord

- er is geen eenduidig criterium om te meten wat het "beste" ontwerp is
- de techniek is heel vrij: elke stap kan verscheidene goede antwoorden bieden
- veel hangt af van de groepsdynamiek

Een voorbeeld v.e. creatieve vraag

+ Je baas heeft op de radio gehoord van het "log4j security exploit" en vraagt een veiligheidsplan. Wat zul je allemaal in dat veiligheidsplan opnemen ?

+ Modelantwoord

+ De context is niet voldoende duidelijk en je moet zelf vragen stellen om die helder te krijgen.

- Wat voor soort systeem is het? Hoe hangt het aan het internet?
- Wat voor soorten risico's loopt het systeem? Hoeveel risico is je baas bereid te lopen? Hoeveel is hij bereid te investeren?

Criteria (ii)

- Levenslang leren
 - + Een professionele software engineer zal zijn leven lang de technische evoluties op de voet moeten volgen.
- * Vele referenties naar boeken, artikels, world-wide web
- * “Engels” als voertaal voor de transparanten

ACHTERGRONDINFORMATIE:

- Je wordt verondersteld zelf selectief met diverse bronnen om te gaan.



Literatuur

Aanbevolen is 1 van onderstaande boeken aandachtig door te nemen (INZICHT)

- + [Ghez02] Fundamentals of software engineering (Second edition), C. Ghezzi, M. Jazayeri, D. Mandrioli, Prentice Hall, 2002.

- > Tijdloos door de nadruk op onderliggende principes, maar daardoor moeilijk.

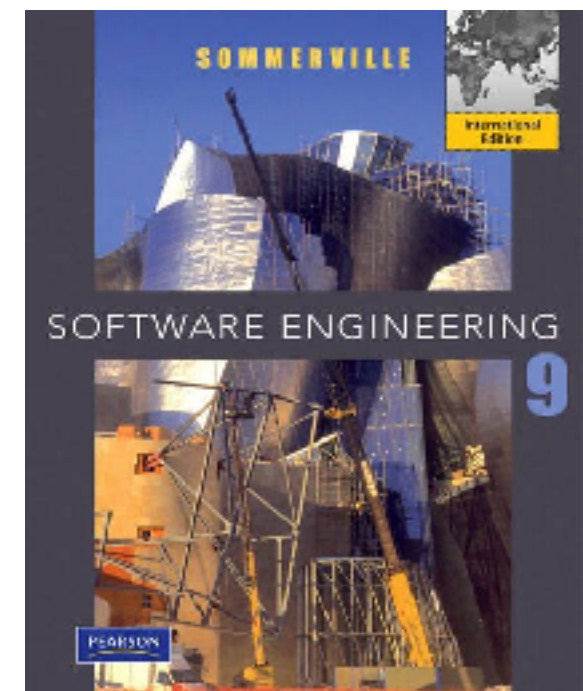
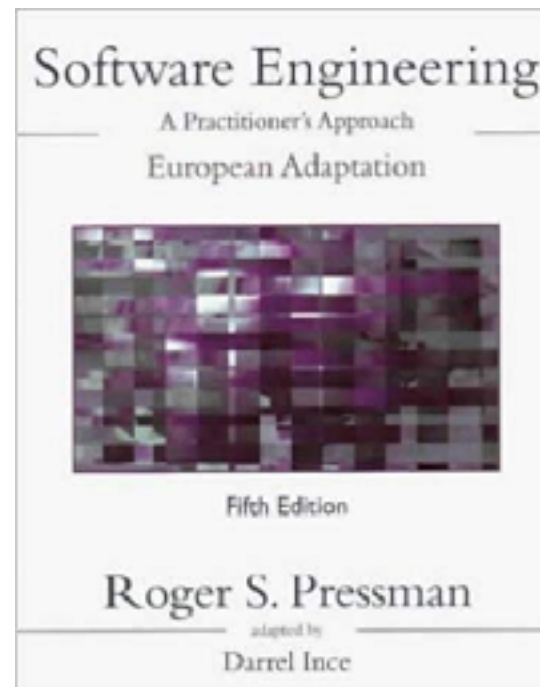
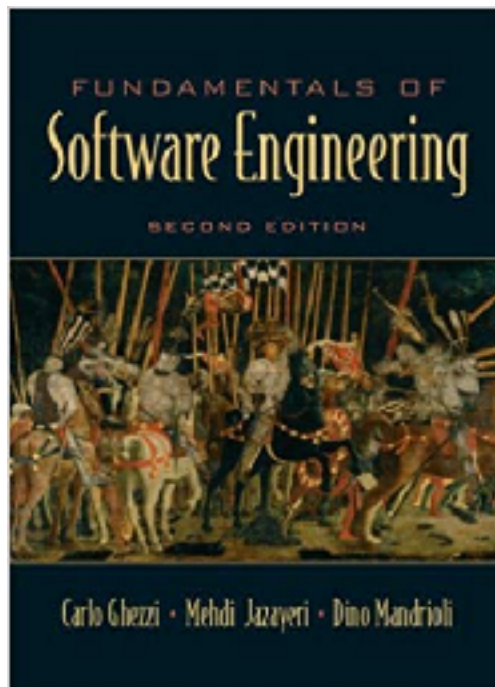
- + [Pres00] Software Engineering — A Practitioner's Approach, R. Pressman (Fifth Edition), Mc-Graw Hill, 2000. (2014 = Eighth edition)

- > Zeer praktisch en zeer diep, maar anderzijds weinig selectief en volumineus (dus duur).

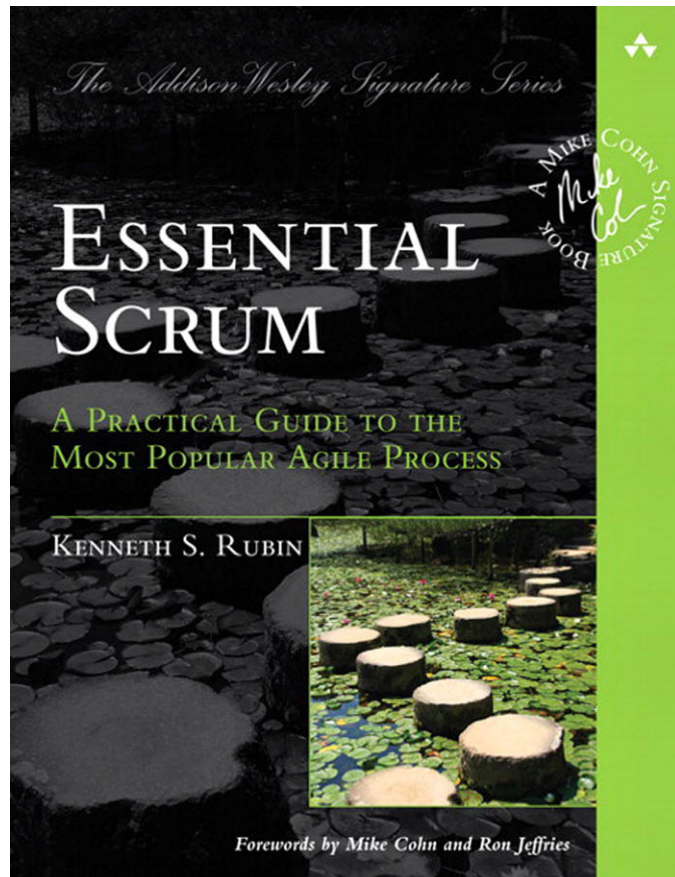
- + [Somm05] Software Engineering (Ninth Edition), I. Sommerville, Addison-Wesley, 2011. (2018 = Tenth edition)

- > Zeer populair, zeer breed en makkelijk leesbaar, maar mist af en toe wat diepgang.

Andere literatuur wordt per hoofdstuk vermeld, incl. referenties op het web.



Literature (Agile)



- [Rubi13] Essential Scrum: A Practical Guide to the most popular agile process. Kenneth S. Rubin. Addison-Wesley, 2013